Cafe Scheduling Optimization Simulation
Final Report
Repository: https://github.com/Baesiann/CS4632_Kenneth_Burke

Kenneth Burke
Department of Computer Science
Kennesaw State University
Kennesaw, Georgia 30144
Email: kburke36@students.kennesaw.edu

Abstract

This project investigates optimal resource allocation (staffing and schedules) in a cafe environment to maximize profits while maintaining acceptable customer service levels. The approach utilizes a discrete-event simulation built with the SimPy library, featuring a doubly-stochastic Poisson process (Cox process) for realistic customer arrivals that model rush periods. Key findings revealed the instability of an initial cost-weight brute-force schedule optimizer, which was prone to pathological behavior by over-weighting labor cost and preferring unrealistically low staffing levels. This was resolved by implementing a threshold-based heuristic scheduler that selects the minimum staff count needed to keep the predicted dropped-customer percentage below 1% of demand, yielding realistic and stable staffing decisions. Main contributions include a flexible Cox arrival generator , a configurable order-table design , and the successful transition to a stable, drop-rate-driven scheduling policy.

Contents

List of Figures

1. Introduction

1.1. Problem Motivation

Service-industry operations, especially in cafes, face the challenge of balancing labor cost against customer satisfaction (wait times and dropped orders) and throughput. Optimal staffing is crucial for maximizing profit.

1.2. Research Questions

The central research question is: How should resources (staffing, skill allocation, schedules) be allocated to maximize profits while maintaining acceptable customer service levels in a cafe environment?

1.3. Project Objectives

The simulation focuses on customer arrivals, order processing, barista scheduling, and short-term profit-relevant metrics. The primary objective was to build a multi-day simulation capable of testing day-level staffing decisions and operational trade-offs.

1.4. Report Organization

This report details the simulation's design and architecture, the challenges encountered with the scheduling algorithm, the final threshold-based scheduling approach, and the resulting analysis and validation of key performance indicators.

2. Background and Literature Review

2.1. Theoretical Foundation (Queuing Theory)

The system models a single-queue, multi-server environment. Metrics like wait time, dropped customers, and average queue size are core concepts derived from queuing theory.

2.2. Related Work

The project builds upon established methods for modeling queuing systems in retail settings using discrete event simulation. Specific literature addresses modeling customer arrivals in service environments like call centers, which is relevant to the chosen arrival model. Simulation-optimization for coffee shops is also a documented approach in operations research.

2.3. Mathematical Models Used

- Arrival Model: Customer arrivals are generated using a doubly-stochastic Poisson process (Cox process) to produce realistic demand bursts, including configurable morning and lunch rush intensities and durations.
- Service Times: Service times for orders are modeled with a mean service time and standard deviation (likely a distribution like Gamma or Exponential, although not explicitly named, the parameters suggest a non-deterministic process).

2.4. Justification for Approach

The discrete-event SimPy simulation approach was chosen for its flexibility in modeling complex, time-dependent processes like queuing, service, and scheduling decisions in detail. The final threshold-based heuristic was adopted because the initial cost-weight optimization proved unstable and non-interpretable due to scaling issues between the penalty terms.

3. Model Design and Architecture

3.1. Conceptual Model Explanation

The simulation models an 8-hour operational day (480 simulated minutes). Customers arrive, join a single queue, and are served by the next available barista, or drop out if their wait exceeds their patience. The staffing level is the main decision variable.

3.2. Entities

- Customers: Have an ID, arrival time, patience, and an order sampled from a configurable distribution.
- Baristas: Modeled as homogeneous SimPy resources (servers); skill differences were initially planned but excluded due to time constraints.
- Orders: Defined by name, price, mean service time, standard deviation, and likelihood.

3.3. Key Design Decisions

- Single Queue: All customers enter a single queue, served by the next free barista.
- Homogeneous Baristas: Simplifies the scheduling problem by focusing solely on staffing count.
- Metric Recording: Timestamps, wait times, service times, drop events, and revenue are recorded per customer into a pandas DataFrame for analysis .

3.4. Implementaion Details

The architecture is modularized. Key modules include:

- `simulation/simulation_runner.py`: Orchestrates single-day and multi-day runs.
- `simulation/customer.py`: Orchestrates doubly-stochastic arrival generator as well as customer attributes.
- `simulation/schedule_manager.py`: Contains the schedule manager logic.

4. Implementation

4.1. Technologies Used

- Language: Python 3.13
- Libraries: SimPy, numpy, pandas, matplotlib, Tkinter
- Development Environment: VS Code / Jupyter notebook (for prototyping)

4.2. Core Algorithms

- Arrival Generation: Doubly-stochastic Poisson process (Cox process).
- Scheduling: Threshold-based heuristic: It tests staffing levels by adding baristas incrementally per day until the realized customer drop rate is below 1%, and if it is incredibly low it decrements a barista for the next day.

4.3. Data Structures

- SimPy Resources: Used to model baristas (servers).
- Pandas DataFrame: Used to record and store all per-customer metrics and aggregated daily KPIs.
- Dictionaries: Holds the orders data containing drink metrics.
- csv/json: Used as an intermediate step between storing simulation results and visualizing the results.

4.4. Challenges Overcome

The major challenge was the unstable scaling of the initial cost-weight schedule manager, which unintentionally overweighted the barista count penalty and resulted in unrealistically low staffing recommendations. This was overcome by replacing it with the threshold-based drop-rate evaluation. Additionally, plumbing the random seed consistently was crucial for test reproducibility.

5. Experimental Setup

5.1. Simulation Parameters

- Setup Parameters
  - Simulation Days: The amount of days the simulation will run for. The higher the amount of days, the more days the simulation will generate arrivals and metrics for.

  - Starting Baristas: The initial amount of baristas scheduled. It affects the first day of the simulation, where the rest of the days the schedule manager will decide how many baristas to schedule.
  - Seed: Can be set to have a consistent stream of random variables.

- Customer Arrival Parameters
  - Baseline Arrival Rate: This is the base arrival rate of customers, a higher rate means more customer arrivals per minute.
  - Morning Rush Intensity: This is how much more volume is expected for a morning rush, it directly affects the baseline arrival rate by creating a peak of baseline arrival + intensity value at hour 2.
  - Lunch Rush Intensity: This is how much more volume is expected for a lunch rush, it directly affects the baseline arrival rate by creating a peak of baseline arrival + intensity value at hour 5.
  - Randomness Intensity: This is how 'random' the arrival distribution will be, the larger the value the more the customers will derive from the expected arrival rates.
  - Morning Rush Duration: This is the time in minutes the morning rush will last for, the rush will start, grow to the peak arrival intensity, and simmer down once the rush ends.
  - Lunch Rush Duration: This is the time in minutes the lunch rush will last for, the rush will start, grow to the peak arrival intensity, and simmer down once the rush ends.

- Order Setup: This parameter is unique in the sense that it is inherently a table of orders. This could change the program entirely as far as revenue and costs go, Each drink can be assigned a price, time to make, a deviation of the time it takes to make, and a probability of it being ordered.

Run ID	**Purpose**	**Parameters Changed**	**Data File**
001	Baseline	defaults	baseline.csv
002	High volume	baseline rate =15	high_vol.csv
003	Many days	simulation days =50	manydays.csv
004	High intensity	"morning =20, lunch =16"	highint.csv
(Various)	Sensitivity Tests	Single parameter changed by +20%	

5.2. Data Collection Methods

Per-customer metrics (timestamps, wait times, service times, etc.) are recorded and exported to CSV/DataFrame . Aggregated daily KPIs (average wait time, revenue, dropped customers, etc.) are computed for post-run analysis.

5.3. Experimental Design

The design involved running stress and nominal scenarios to observe the stability of metrics and the responsiveness of the scheduler. Sensitivity testing was performed by perturbing a single parameter by +20% and measuring the percent change in outputs over a 5-day sample.

6. Results and Analysis

6.1. Sensitivity Analysis Findings

  - Randomness Intensity: Showed high sensitivity, with increased randomness causing large increases in wait time, drops, and queue sizes.
  - Baseline Arrival: Produced mixed sensitivity; some runs showed the expected increase in arrivals and wait, while others showed small or negative changes (attributed to sampling variability or seeding issues).
  - Throughput Reward: Produced near-zero sensitivity, as the initial cost-based scheduler often ignored this reward relative to the labor cost penalty.

6.2. Scenario Comparisons

Scenario testing confirmed that increased arrival intensity and sharper rush patterns cause higher average wait times and queue congestion, while removing rushes produces a more uniform flow.

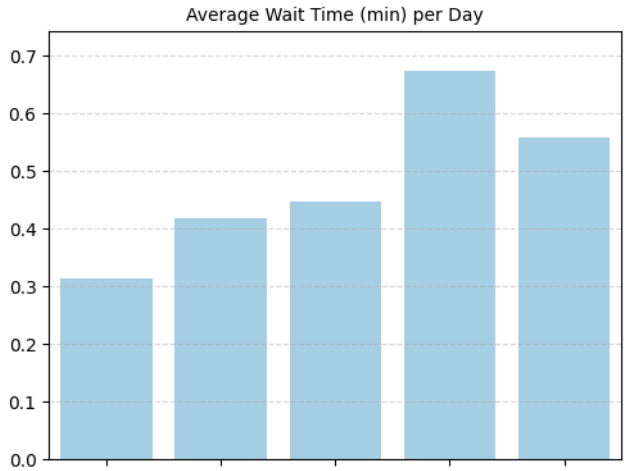
6.3. Statistical Analysis

Baseline runs (5 replications, 6 days each) yielded the following statistical summary:

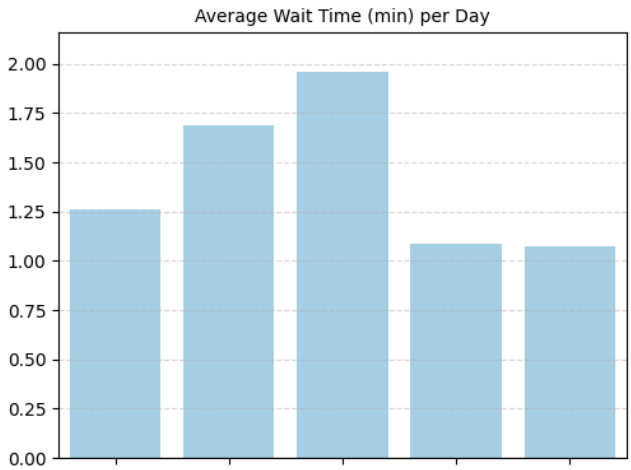
- Average Wait Time (min): Mean 2.938 (Std Dev 0.438)
- Total Revenue ($): Mean 369.8 (Std Dev 33.502)
- Dropped Customers (count): Mean 9.267 (Std Dev 2.987)

6.4. Visualizations with Explanations

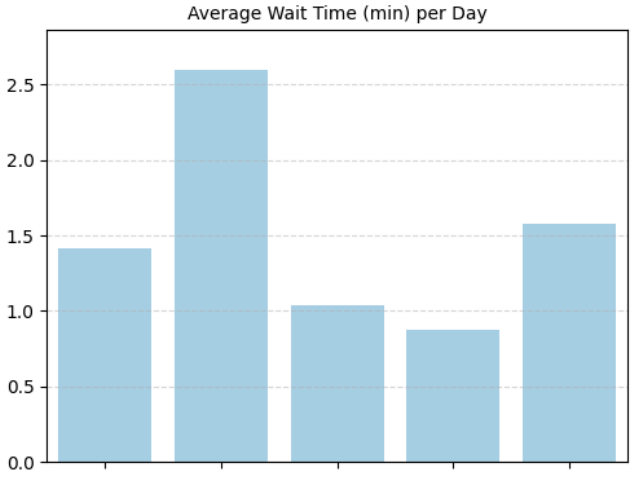
Wait Time vs. Arrival Profiles: Bar plots confirm that high volume and high-intensity rushes lead to higher average wait times compared to baseline or no-rush scenarios. * Staffing Behavior: The final threshold-based scheduler consistently selected two baristas per day in the baseline run, indicating stable demand and successful drop-rate suppression below the 1% threshold. * Daily Performance Metrics: Plots demonstrate that once the scheduler stabilized (after the first day), dropped customers remained near zero, throughput remained consistent due to stable staffing, and wait time/queue size remained controlled over the simulated week.



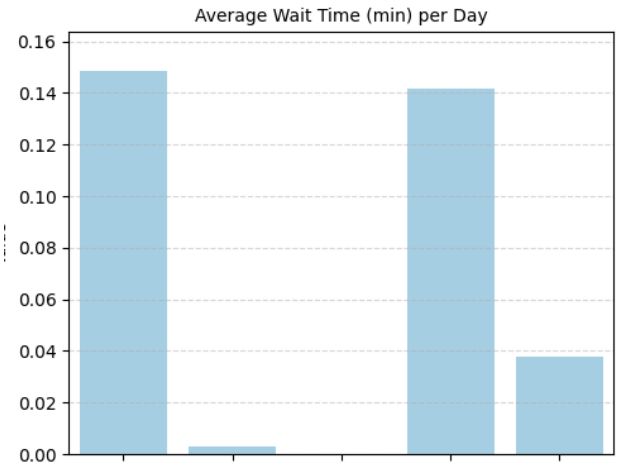
(a) Baseline arrival profile



(b) High volume (baseline rate = 15)



(c) High intensity rushes



(d) No rush (both intensities = 0)

Figure 1: Average wait time under different arrival profiles

These variations confirm that increased arrival intensity and sharper rush patterns cause higher average wait times and queue congestion, while removing rushes produces a more uniform flow.

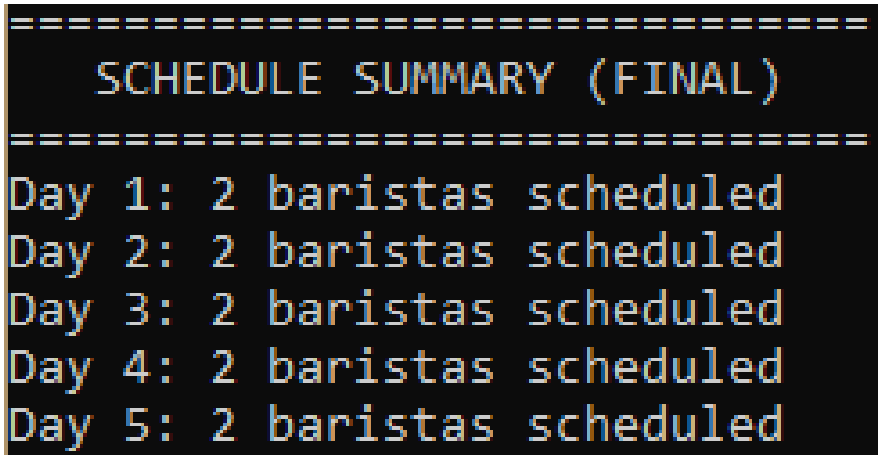
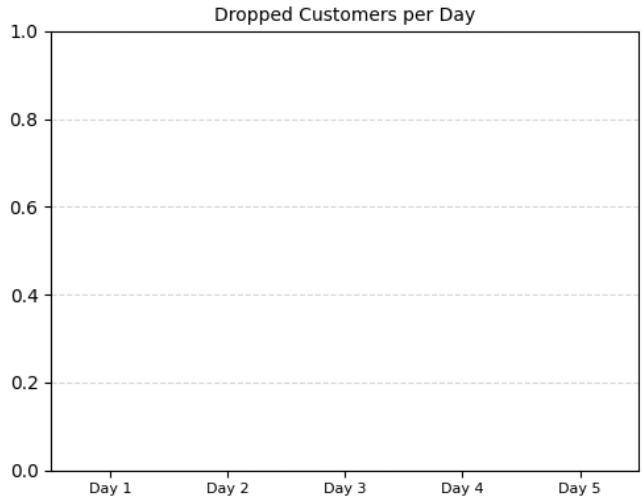
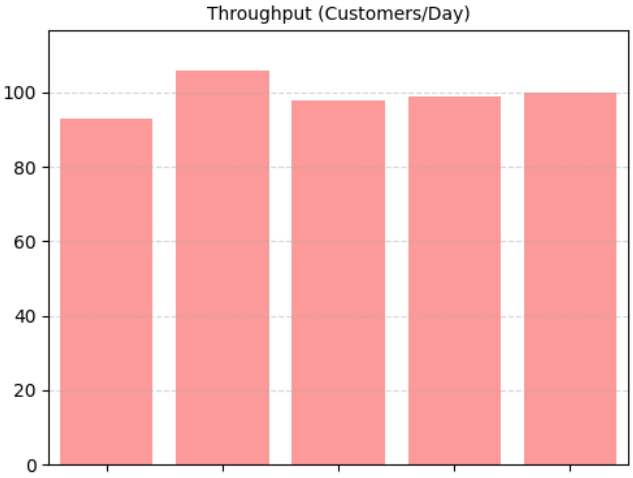


Figure 2: Number of baristas scheduled per day under the final threshold-based scheduler

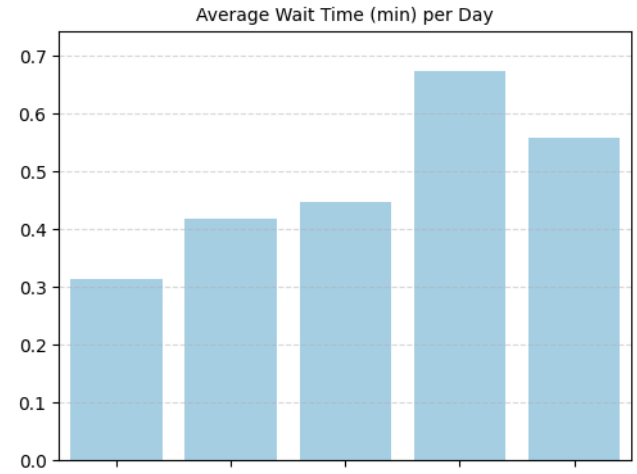
The threshold-based scheduler evaluates increasing staffing levels until the predicted dropped-customer percentage falls below 1% of the previous day's arrivals. In this run, the model consistently selected two baristas each day, indicating stable demand and successful drop-rate suppression.



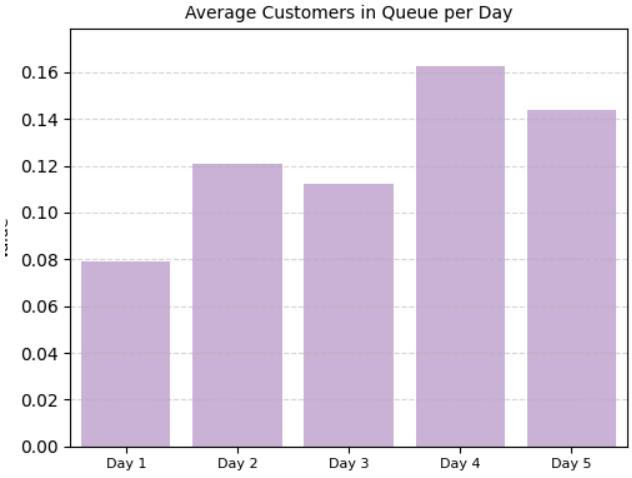
(a) Dropped customers per day



(b) Throughput per day



(c) Average wait time per day



(d) Average queue length per day

Figure 3: Daily operational performance under the final scheduling system

These plots collectively demonstrate that:

- Dropped customers remain near zero after the first simulated day
- Throughput stays consistent due to stable staffing
- Average wait time and queue size remain controlled
- Performance does not degrade over the simulated week

7. Validation and Verification

7.1. Validation Methods Used

- Face Validity: Wait and queue patterns react intuitively to changes in arrival intensity and rush structure (e.g., higher intensity yields higher average wait times).
- Parameter Validation: The orders table maps sensibly to revenue/service-time behavior (e.g., longer, more complex drinks are more expensive and introduce more variance).

7.2. Evidence of Correctness

The major evidence of correctness is the successful transition to the threshold-based scheduling heuristic, which consistently produces reasonable staffing adjustments tied directly to operational performance. This eliminated the

pathological behavior of preferring unrealistically low staffing.

7.3. Limitations Acknowledged

- Critical Failure: The initial schedule manager's behavior under heavy load was a critical validation failure.
- Parameter Plumbing: Some sensitivity tests returned counter-intuitive results, pointing to potential sampling noise or implementation plumbing issues in parameter passage.
- Reactive Scheduler: The current threshold-based scheduler is reactive, relying entirely on the previous day's demand as a predictor, making it potentially brittle under highly variable demand.

8. Discussion

8.1. Interpretation of Results

The results confirm that the simulation accurately captures intuitive queue dynamics: increasing arrival intensity drastically increases wait times and drops. The final scheduling system successfully meets its primary goal of suppressing dropped customers by consistently adding the minimum staff required to pass the 1% drop threshold.

8.2. Answered Research Questions

The project demonstrated that staffing resources can be allocated via a simple, threshold-based heuristic tied directly to customer service outcomes (dropped customers) to maintain acceptable service levels.

9. Practical Implications

The work revealed how sensitive cafe performance is to small changes in barista capacity. It provides a tool for policy trade-offs focused on day-level staffing decisions.

9.1. Surprising Findings

The most surprising behavior was the instability of the initial weighted cost function, where labor cost was unintentionally over-weighted, causing the optimizer to prefer a single barista even when lost revenue from drops was substantial.

10. Conclusion and Future Work

10.1. Summary of Contributions

The project successfully delivered a flexible discrete-event simulator with a realistic Cox arrival model, configurable menu, data pipeline, and a functional GUI. The successful implementation of the threshold-driven scheduler greatly improved the realism and stability of the decision-making.

10.2. Lessons Learned

The primary lesson was that complex, multi-variable cost functions require extensive normalization and tuning, and often a simpler, directly interpretable heuristic (like the drop-rate threshold) is superior for operational decision-making.

10.3. Potential Improvements

- Switch the scheduler to forward-simulation evaluation of candidate schedules to minimize expected cost across demand scenarios.
- Improve GUI responsiveness (e.g., defer plotting or use background threads).

10.4. Future Research Directions

Future work should focus on:

- Integrating barista heterogeneity (skills and task routing) and fatigue/learning effects.
- Harden the arrival model plumbing with unit tests and increase statistical rigor via more replications and confidence intervals.

References

[1] C. Sutton and M. I. Jordan, "Bayesian inference for queueing networks and modeling of internet services," *The Annals of Applied Statistics*, vol. 5, no. 1, Mar. 2011.

[2] D. Buzali, S. Elizondo, S. Muñiz, and O. Sánchez, "Simulation-optimization of a coffee shop in business district: A case study of Starbucks in Mexico City," Proceedings of the International Conference on Industrial Engineering and Operations Management, May 2024.

[3] R. Ibrahim, H. Ye, P. L'Ecuyer, and H. Shen, "Modeling and forecasting call center arrivals: A literature survey and a case study," *International Journal of Forecasting*, vol. 32, no. 3, 2016.

[4] L. R. de Groot and A. Hübl, "Modeling queuing system in retail using discrete event simulation," Proceedings of the 2021 Winter Simulation Conference (WSC).

[5] Y. Zhang, "Simulation and analysis of queueing system," Master's Thesis, KTH Royal Institute of Technology, 2019.